

METHOD AND APPARATUS FOR IMPROVING MESSAGE
AVAILABILITY IN A SUBSYSTEM WHICH SUPPORTS
SHARED MESSAGE QUEUES

5 CROSS REFERENCES TO RELATED PATENT APPLICATIONS

10 This application claims priority and all other
benefits under 35 U.S.C 120 of prior filed co-pending US
provisional patent application US Serial No. 60/220,685,
filed July 25, 2000, which is incorporated herein by
reference.

15 This application is related to the subject matter of
the following co-pending patent applications, each of which
is assigned to the same assignee as this application and
each of which is incorporated herein by reference:

20 D. A. Elko et al., US Serial No. 09/677,339, filed
October 2, 2000, entitled "METHOD AND APPARATUS FOR
PROCESSING A LIST STRUCTURE" (IBM docket POU920000043);

25 D. A. Elko et al, US Serial No. 09/677,341, filed
October 2, 2000, entitled "METHOD AND APPARATUS FOR
IMPLEMENTING A SHARED MESSAGE QUEUE USING A LIST STRUCTURE"
(IBM Docket POU920000042);

P. Kettley et al., US Serial No. 09/605,589, filed June 28, 2000, entitled "METHOD AND APPARATUS FOR OPERATING A COMPUTER SYSTEM TO ENABLE A RESTART" (IBM docket GB920000031);

5

P. Kettley et al., US Serial No. 60/219,889, filed July 21, 2000, entitled "IMPLEMENTING MQI INDEXED QUEUE SUPPORT USING COUPLING FACILITY LIST STRUCTURES" (IBM docket GB920000033).

10

FIELD OF INVENTION

The present invention relates to methods and apparatus for recovery from failures affecting a resource manager within a data processing network, and has particular applicability to the field of communicating message data between application programs via shared message queues and to the resolution of a failed resource manager's units of work by other active resource managers to improve message availability.

15

20

25

30

35

40

45

50

BACKGROUND OF THE INVENTION

In the communication through a computer network of message data between application programs it is known to transmit the messages by means of resource managers such as queue managers which interface to the application programs

25

through a message queueing interface that is invoked by the application programs. In contemporary data processing environments, it is commonplace for the computer network to connect a client application program that has a task to be performed to one or more transaction-oriented resource manager programs that will undertake the task. In this environment, the client queues an input message through a queueing subsystem to a target system to request processing and when the processing is complete a return message is queued for transmission back to the client.

US Patent 5 452 430 describes a data processing system for the storage of persistent and non-persistent data in a queue and a method for the storage of data which is required to survive a system failure. The method involves receiving persistent and non-persistent data to be stored in a queue, then marking the data in time sequence order before storing the persistent data in a first set of data pages and the non-persistent data in a second set of data pages. Upon receiving a request to remove data from the queue, both the first and second pages are checked and the data is removed from the queue in time sequence order. A

log is created to enable recovery in the event of failure and restart of the queue.

It is also known from US Patents 5 797 005 and 5 887 168 to provide a system allowing messages to be processed by any of a plurality of data processing systems in a data processing environment. A shared queue is provided to store incoming messages from message queuing subsystems for processing by one of the plurality of data processing systems. A common queue server receives and queues the messages from a subsystem onto the shared queue so that they can be retrieved by a system having available capacity to process the messages. A system having available capacity retrieves the queued message, performs the necessary processing and places an appropriate response message back on the shared queue. Thus, the shared queue stores messages sent in either direction between clients requesting processing and the data processing systems that perform the processing. Because the messages are enqueued onto the shared queue, the messages can be processed by an application running on any of a plurality of systems having access to the queue. Automatic workload sharing and processing redundancy is provided by this arrangement. If a particular application that is processing a message fails, another application can retrieve that message from the shared queue and perform the processing without the client

having to wait for the original application to be restarted.

SUMMARY OF THE INVENTION

5

It is the aim of the present invention to improve recovery from a connection failure between a resource manager such as a queuing subsystem and a shared resource such as a shared queue, such failure being caused either by communications link failure, or failure of the computer system or of a computer program comprising the resource manager.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995

In a first aspect of the present invention, there is provided a method for recovering from failures affecting a resource manager within a group of resource managers, wherein the resource managers within the group have access to a shared resource via which remote resource managers communicate with the resource managers within the group, the shared resource including data storage structures to which resource managers within said group connect to send and receive communications, the method comprising:

storing, within a first data storage structure of the shared resource, unit of work descriptors for operations

performed in relation to said shared resource by the
resource managers in said group;

5 sending a notification of a connection failure between
a second data storage structure of the shared resource and
a first resource manager within said group, the
notification being sent to the remaining resource managers
within the group which are connected to the second data
storage structure;

10 one or more of said remaining resource managers
accessing said first data storage structure and analysing
the unit of work descriptors to identify the units of work
relating to the second data storage structure that were
15 being performed by the first resource manager when the
connection failure occurred; and

20 said one or more remaining resource managers
recovering the identified units of work.

25 In one embodiment, there is provided a method of
communicating information relating the state of units of
work and the messages which form part of the unit of work
between message queuing subsystems coupled together through
a coupling facility, the method comprising: communicating
the message data in at least one shared queue between the

message queuing subsystems by means of data structures
contained in the coupling facility; notifying a connection
failure between a queuing subsystem and the coupling
facility data structure containing the shared queue, the
notification being provided to the remaining queuing
subsystems in communication with the coupling facility;
including, within the data structure containing the shared
queue, information describing which shared queues within
the structure are in use from a particular queuing
subsystem; including within the said data structures an
administrative structure listing unit of work descriptors
describing operations performed by the queuing subsystems
on those shared queues which are contained within a certain
coupling facility data structure; including with the
message data stored in the coupling facility data
structures, a key to enable retrieval of the message data,
one range of key values identifying a message as committed,
a further range of key values identifying a message as
uncommitted, and a third range of key values identifying
the message state as indeterminate; after a connection
failure, employing the said remaining queuing subsystems in
parallel to interrogate the listed work descriptors so as
to identify and to share between them the recovery of units
of work active in the subsystem whose connection to the
coupling facility has failed, and employing each of the
said remaining subsystems to recover its share of the units

of work active in the queuing subsystem whose connection to the coupling facility has failed.

After all unit of work descriptors are processed, a
5 second phase of recovery is initiated in which the said remaining subsystems find all inflight messages read and written by the failing queuing subsystem and roll them back. These inflight messages are found without reference to the failed queuing subsystem's logs.

10 In a second aspect of the invention, there is provided a method for recovering from failures affecting a resource manager within a group of resource managers, wherein the resource managers within the group have access to a shared
15 resource, the shared resource including data storage structures to which resource managers within said group connect to perform operations in relation to data held in said shared resource, the method comprising:

20 storing, within a first data storage structure of the shared resource, unit of work descriptors for operations performed by the resource managers in said group in relation to data held in said shared resource;

25 sending a notification of a connection failure between a second data storage structure of the shared resource and

a first resource manager within said group, the notification being sent to the remaining resource managers within the group which are connected to the second data storage structure;

5

one or more of said remaining resource managers accessing said first data storage structure and analysing the unit of work descriptors to identify the units of work relating to the second data storage structure that were being performed by the first resource manager when the connection failure occurred; and

10

said one or more remaining resource managers recovering the identified units of work.

15

In a third aspect of the invention, there is provided a distributed data processing system including:

20

a plurality of resource managers;

20

a shared access resource including data storage structures to which the resource managers connect to send

and receive communications to and from remote resource managers, the shared access resource including:

means for storing, within a first data storage structure of the shared resource, unit of work descriptors for operations performed in relation to said shared resource by the resource managers in said plurality; and

means for sending a notification of a connection failure between a second data storage structure of the shared resource and a first resource manager within said plurality, the notification being sent to the remaining resource managers within the plurality which are connected to the second data storage structure;

wherein said remaining resource managers include:

means for accessing said first data storage structure and analysing the unit of work descriptors to identify the units of work relating to the second data storage structure that were being performed by the first resource manager when the connection failure occurred; and

means for recovering the identified units of work.

In a further aspect of the invention, there is provided a computer program product comprising program code recorded on a machine-readable recording medium, the program code comprising the following set of components:

5

a plurality of resource managers;

a shared access resource manager including program code for managing storage and retrieval of data within data storage structures to which the resource managers connect to send and receive communications to and from remote resource managers, the shared access resource manager including:

10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198

wherein said remaining resource managers include:

means for accessing said first data storage structure
5 and analysing the unit of work descriptors to identify
the units of work relating to the second data storage
structure that were being performed by the first resource
manager when the connection failure occurred; and
means for recovering the identified units of work.

10 According to another embodiment of the invention,
there is provided a communication system to communicate
information relating the state of units of work and the
messages that form part of the unit of work between message
15 queueing subsystems, the system comprising: a coupling
facility to communicate the message data between the
message queueing subsystems in at least one shared queue by
means of data structures included within the coupling
facility; means to notify a connection failure between a
20 queueing subsystem and the coupling facility data structure
containing the shared queue, the notification being
provided to the remaining queueing subsystems in
communication with the coupling facility; the data
structure containing the shared queue including information
25 describing which shared queues within the structure are in
use from a particular queueing subsystem; an administrative

structure within the data structures listing unit of work descriptors describing operations performed by the queuing subsystems on those shared queues which are contained within a certain coupling facility data structure; the message data stored in the coupling facility data structures including a key to enable retrieval of the message data, one range of key values identifying a message as committed, a further range of key values identifying a message as uncommitted, and a third range of key values identifying the message data as indeterminate; means operative, after a connection failure, to employ the said remaining queueing subsystems in parallel to interrogate the listed work descriptors so as to identify and to share between them the recovery of units of work active in the subsystem whose connection to the coupling facility has failed and to employ each of the said remaining subsystems to recover its share of the units of work active in the queueing subsystem whose connection to the coupling facility has failed.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described in more detail, by way of example, with reference to the accompanying drawings in which;

Figure 1 shows a message queue shared between a plurality of application programs;

Figure 2 shows a message queuing system according to the present invention to communicate message data between application programs connected to the system;

Figure 3 shows an application structure included within the system of Figure 2;

Figure 4 shows an administrative structure included within the system of Figure 2; and

Figure 5 shows steps performed in recovering units of work active in a failed connection to a shared queue.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention is directed to methods and systems to communicate message data between application programs connected to a system and to recover units of

work, consisting of messages read and written, owned by the failing queuing subsystem in parallel by surviving queuing subsystems without access to logs maintained by the failed queuing subsystem.

5

Figure 1 shows a shared queue 10 which receives messages 11, 12, 13 put onto the queue by application programs 14, 15, 16. Programs 17, 18 and 19 get messages from the shared queue 10. The programs 14, 15 and 16 represent client application programs including instructions for performing client-related tasks. When a client program has a transaction to be processed it places a message on the shared queue 10 to be retrieved by one of the programs 17, 18 and 19. When one of the programs 17, 18 and 19 gets a message, it processes the message and places a response on the shared queue 10. Each message remains on the queue until it is retrieved. It will be understood that the diagram of Figure 1 has been simplified for ease of explanation. In practice, a network of connections could be used to communicate messages between the programs requiring to communicate with one another and whilst only one shared queue 10 has been shown for simplicity of explanation, in practice a number of such shared queues are used to pass messages between the communicating programs.

10

15

20

25

Referring now to Figure 2, message queuing subsystems 20 and 21 are provided to interface the programs 14, 15, 16, 17, 18 and 19 to shared queue 10. Whilst only two message queuing subsystems have been shown in Figure 2, it will be apparent to one skilled in the art that the number of subsystems may vary. A coupling facility 22 is provided to couple the message queuing subsystems 20 and 21. A coupling facility manager component 23 is included in each message queuing subsystem to interface to the coupling facility 22 through the use of services provided by the operating system.

Each application program 14, 15, 16, 17, 18 and 19 operates on a computer platform comprising a computer and an operating system. The computer typically includes various hardware components, such as one or more central processing units (CPU), a random access memory (RAM) and an input/output (I/O) interface. The message queue subsystems 20 and 21 also run on a computer platform comprising a computer and an operating system and connected to interface to the application programs 14, 15, 16, 17, 18 and 19 either locally or through a communication network which may be implemented as a local area network (LAN) or a wide area network (WAN) for example.

The coupling facility 22 runs on a further computer platform including a computer and an operating system. A

computer program product comprising instructions stored on a computer readable medium enables the computer to execute the features of the coupling facility as will be discussed below. The coupling facility sets aside an area of storage to hold the messages in the shared queue 10 on their way from one program to another. The communications link between the coupling facility 22 and the connecting computer systems running the message queuing subsystems 20, 21 is managed by a component (referred to as XES) of the operating system of each of the connecting computer systems. The message queuing subsystems 20,21 and the coupling facility 22 may all run on the same computer or same computer platform.

The XES operating system component provides functions that allow the message queuing subsystems 20 and 21 to allocate data structures within the coupling facility 22 and to connect to and manipulate these data structures. The XES component also provides status information concerning the status of concurrent connections to the same data structures within the coupling facility 22. The coupling facility managers 23 move data between the coupling facility and the message queuing subsystems 20 and 21. Through the use of the XES services, the coupling facility manager component of the queuing subsystem connects to a single administrative structure 25 used for communicating

unit of work information between the queuing subsystems,
and one or more application structures 24 to hold the
application message data. Each queuing subsystem sharing
message queues will have a connection to the administrative
5 structure and connections to the application structures
which hold message data associated with the shared queues
being operated on by applications attached via that queuing
subsystem. These connections are indicated by the arrows
in Figure 2.

10 Referring now to Figure 3, the application structure
24 will be described. The application structure contains an
array of list headers 26. Each of the list headers 26 may
have anchored to it zero or more list entries that contain
15 message data put by an application interfacing with a
queuing subsystem to a shared queue. The list entries each
have a key and the list entries are listed under each list
header in the order of their keys. The construction of the
keys and mechanism for storing and retrieving messages
20 using these is described in commonly assigned US patent
application Serial No. 09/677,341, which is incorporated
herein by reference (IBM docket POU920000042). USSN
09/677,341 describes how keys represent committed,
uncommitted, or 'indeterminate' state messages. The keys
25 also specify the queue manager (queueing subsystem) which
put the message on the queue and/or of the queue manager

which performed a get operation to retrieve the message,
and a timestamp of when the message arrived on the queue.
The lowest keyed entry in a list is at the head of its list
and the highest keyed entry at the tail of its list. A
5 first set 27 of the list headers is a set of shared queue
list headers corresponding respectively to the shared
queues 10. A second set 28 of the list headers will be
referred to as uncommitted get queue list headers. Each
queuing subsystem connected to a coupling facility data
10 structure 22 has its own unique uncommitted get queue.
Making entries to the uncommitted get queue lists is
explained below.

15 A list header 29 at the top of the array of list
headers has a list of data representing a list header
information map 30. For each message queuing subsystem, the
information map 30 identifies the shared queue list headers
in the array 27 currently being used by that message
queuing subsystem. Each queuing subsystem has its own list
20 header information map (LHIM). The LHIM is bound to a
specific queuing subsystem based on the LHIM's list entry
key. There is a bit string in the LHIM and a one to one
correspondence between bits in the bitstring and list
headers in the data structure such that, if bit 1 is set
25 on, then the queuing subsystem which owns the LHIM has an
interest in the shared queue which maps to list header 1

and so on. The coupling facility manager 23 of each message queuing subsystem maintains the entries in the information map 30 for that message queuing subsystem. When a program puts a message onto a shared queue, the coupling facility manager 23 interfacing to that program moves the message data from private storage in the message queuing subsystem to a list entry with a key value in the uncommitted key range in the coupling facility data structure and associates it with the list header in the array 27 identified with that shared queue. When a program gets a message from a shared queue, the coupling facility manager 23 finds the list entry with the lowest key value in the committed range associated with the list header in the array 27 corresponding to the shared queue and copies the message data into private storage in the message queuing subsystem. At the same time, the list entry is dissociated from the list from which it was copied and reassociated with the uncommitted get queue list corresponding to the message queuing subsystem, and the list entry key value is changed to a value in the uncommitted range. Thus messages forming part of an uncommitted unit of work have list entry key values in the uncommitted range.

So far, only simple put and get operations have been described. In practice, an application program will perform a series of linked operations to perform a unit of work. A

unit of work is a term of art that refers to a recoverable series of operations performed by an application between two points of consistency. A unit of work begins when a transaction starts or after a user-requested

5 synchronisation point (syncpoint). It ends either at a user-requested syncpoint or at the end of a transaction. A unit of work may involve an application performing a set of operations such as getting a message from a queue, making a database entry based on the contents of the message and

10 putting a different message back onto a queue indicating the results of the database operation. In a cash machine application for example, the cash machine may send a message indicating the withdrawal of a particular amount of cash, the computer at the bank will retrieve the message,

15 debit the appropriate account and send back the current balance. In real life, a program will want either all or none of these steps to occur. So if the database update fails, for example, the original request message should be put back into the queue so that another program may try

20 again.

When the thread of operations that are executed in a unit of work reaches a synchronisation point, they can either be done (known in the art as committed) or undone

25 (known in the art as backed-out or aborted). When an application is part way through the thread of operations in

a unit of work, it is known as inflight so that if it abnormally terminates, the message queuing subsystem can detect this and is able to back-out the updates made by the application. The sequence of operations may then be retried from the beginning by another program.

A unit of work may be also be in a fourth state known in the art as 'indoubt'. The 'indoubt' state is associated with two phase commit protocols and indicates that the queuing subsystem is unable to make the decision as to whether the unit of work should be committed or backed out as the unit of work (typically) involves other resource managers, such as a database, and must be coordinated by an external syncpoint manager.

The present invention aims to provide sufficient information in the shared storage within the coupling facility 22 that if one of the message queuing subsystems in one computer fails, another message queuing subsystem in another computer is able to back-out or commit the units of work in progress on the failed computer at the time of failure. Figure 4 shows the administrative structure 25 that is used to provide the shared storage for information about operations performed on shared queues in units of work. For units of work progressing beyond the inflight state, a unit of work descriptor is built for each coupling

facility list structure accessed. The unit of work descriptor (UOWD) identifies the list entries containing message data which have been affected by this unit of work in the corresponding coupling facility list structure, and the operation, either a get or put, which has been performed. For example, if the unit of work has accessed multiple coupling facility structures (because for example shared queue 1 maps to structure 1 and shared queue maps to structure 2 and the application did a put to each queue) multiple unit of work descriptors are written - one (or more) for each coupling facility structure accessed by the unit of work. In this case, each unit of work descriptor points to a summary list entry that identifies the set of coupling facility structures the unit of work accessed. For a commit operation, inflight messages put in the unit of work have their list entry key value modified so that it has a value in the committed range, and inflight messages touched by a get operation are deleted from the uncommitted get list header. For a backout operation, inflight messages put in the unit of work are deleted, and inflight messages got in the unit of work are moved back from the uncommitted get queue to the list header from which they had been gotten, and the key value of the list header changed from uncommitted back to a value in the committed range. It is clear that with information about which list entries have been affected in the unit of work, and the

operation (put or get) performed, these operations can be performed by a different queuing subsystem after a failure. As an optimization, no information is kept in the administrative structure about inflight units of work.

5

The administrative structure 25 has an array 31 of list headers. Each of the list headers may have a list of data entries associated with it in the manner already described with reference to the application structure of Figure 3. The array 31 also includes unit of work list headers 34. Each message queuing subsystem is assigned a unit of work list header 34. The UOWD is written into the administrative structure just before the syncpoint operation and associated with the unit of work list header 34 of the queueing subsystem on which the unit of work has been performed. After the syncpoint operation has been performed the UOWD is no longer required and is deleted.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500

Referring now to Figure 5, the operations of the message queuing subsystems 20 and 21 and the coupling facility 22 will be described in relation to the recovery of units of work after the failure of a connection between the queuing subsystem and the application structure of Figure 3. In a first step 35, a connection to an application structure in a coupling facility fails. The XES program issues in step 36 a notification of the failure to

20

25

any remaining connectors to the same application structure. The remaining connectors may be referred to as peer connectors.

5 Each peer, on receiving a notification from XES that a connection failure has occurred, attempts to start peer recovery by using XES services to broadcast to all peers an instruction to start peer recovery. XES will ensure that only one of these requests is successful and that all peers receive such a request. Peer recovery takes place in two phases, in the first phase units of recovery for the failed connection which are represented by UOWDs associated with the unit of work list header for the queuing subsystem whose connection has failed are recovered. In the second phase, inflight units of work are recovered. Again XES services are used to ensure that phase 1 has completed in all peers before phase 2 can start in any peers.

10
15
20 Each peer receives in step 37 a request to start peer recovery for the failed connector for the application structure. The process enters a phase 1 of recovery in which the unit of work list header for the failed connector is established in step 38 from information contained in the connection identifier for the failed connection which is notified to the surviving connectors by XES. In step 39 the peers work in parallel through the units of work

descriptors in the list for the failed connector and each selects a unit of work for recovery (when a peer finds there are no further units of work to recover, it has completed phase 1). The ownership and recovery of an individual unit of work is effected through the version number mechanism described in the attached design documentation. In step 40 each peer recovers the unit of work that it has selected in its entirety on its own and then makes a selection of the next unit of work that is available for recovery (i.e. not selected and owned by another peer). The recovery of a unit of work involves completing the commit for units of work marked in-commit, backing out units of work marked in-abort, or moving to an 'indeterminate' key range, the messages in a unit of work marked as in-doubt. The peers thus co-operate in parallel together, selecting a single unit of work each at a time for recovery and working down the list of unit of work descriptors in the list. When there are no further unit of work descriptors associated with the queuing subsystem whose connection has failed left to process, a peer indicates that it has finished phase 1 of recovery. In step 41 each peer issues a confirmation to the XES program that it has completed phase 1 of recovery and in step 42 the XES program issues to all the peers a notification when all the peers have confirmed the completion of phase 1 of recovery so that the peers can all start phase 2.

In a second phase of recovery, the recovery work to be performed is BACK-OUT of inflight activity. In step 43, each peer refers to the information map 30 in the application structure to find the shared queue list headers that may have inflight activity. In step 44, each peer selects a list header for inflight recovery and prevents other peers from also using it through the use of an ENQ (serialization mechanism). Inflight recovery involves deleting list entries with a key in the uncommitted keyrange which were put by the queuing subsystem whose connection to the coupling facility has failed, and finding list entries with a key in the uncommitted key range on the uncommitted get queue and which were originally associated with the shared queue being processed and moving them back to that list header, at the same time, changing the key value of the list entry to be in the committed key range. In step 45, each peer recovers the entries under the selected list header entirely on its own and then deals with the next available list header. The peers co-operate together in parallel to each recover a list header at a time. When a peer has recovered a list header, it updates the entry in the information map 30 to reset the interest of the failed connection in that list header. At the end of phase 2, the information map 30 shows that the failed connection has no interest in any of the list headers. No

further recovery work can be performed by peers, although on restart of the failed queueing subsystem there may be indoubt messages whose list entries had been moved to keys in the 'indeterminate' key range by peer recovery and which can be either committed or backed out after reference to the restarting queuing subsystem's log or the external syncpoint manager. Each peer notifies XES that it has completed phase 2.

Because of the parallel nature of the recovery performed by the peer connectors, it is possible to undertake the recovery of more than one failed connection to a shared queue. Thus if a connection A fails and peer connections B and C survive, the peer connections B and C operate in parallel to recover the connection A. If peer B should fail during the recovery of A, the remaining peer C will complete the recovery of A and then proceed to recover peer B, including any recovery which B was doing for A as indicated by B's ownership of UOWDs on A's unit of work list header.

What has been described above is the processing performed by surviving queueing subsystems with respect to a single application structure. In general, the failed queueing subsystem may have been connected to multiple application structures, for example structure 1 and 2. If a

surviving queuing subsystem is also connected to both structures 1 and 2, it receives two notifications from XES, one for structure 1 and another for structure 2. The surviving queuing subsystem in this case is able to perform the recovery processing (as described above) in parallel for the shared queues that map to structure 1 and structure 2.

This results in the ability to recover a multiplicity of shared queues in parallel, the set of shared queues distributed over a set of application structures provided a surviving peer is connected to the same set (or a subset) of the structures to which the failing queuing subsystem had a connection.

What has been described is a method and system for communicating unit of work information between a plurality of message queuing subsystems. The unit of work state information being provided either through the use of keys, or explicitly through Unit of Work Descriptors stored in the administrative structure. This communication of unit of work information means that in the event of failure of a message queuing subsystem or its communication with the shared queues, other message queueing subsystems can progress the active units of work without reference to the failing subsystem's log data or waiting for the failed

queuing subsystem to restart, so improving message availability. The maintenance of the list header information map means that the queuing subsystem performing the recovery can rapidly find any inflight units of work in progress at the time of failure, and back them out without the overhead of explicitly maintaining in a shared place the list of messages in the unit of work. A plurality of message queuing subsystems 20, 21 interface to the application programs and are coupled together through one or more coupling facilities 22. The functioning of the subsystems 20 and 21 and the coupling facility 22 are illustrative of the functions to be performed in communicating message data between the application programs to which the subsystems interface. The message data is communicated in shared queues between the message queuing subsystems by means of the data structures 24 contained in the coupling facility. The application structure 24 lists shared queues to which each message queuing subsystem is connected and the administrative structure 25 lists unit of work descriptors describing operations performed by the queuing subsystems on the shared queues for units of work which have reached a syncpoint operation. A connection failure between a queuing subsystem and a shared queue is provided to the remaining queuing subsystems connected to the shared queue. The remaining queuing subsystems operate in parallel to interrogate the listed work descriptors so

as to identify and to share between them the units of work active in the failed connection, and each of the said remaining subsystems recovers its share of the units of work active in the failed connection.

5

The solution described above includes using a structure interest map in a second phase of recovery, to identify the Coupling Facility list structures to which a resource manager (queuing subsystem) was connected. The second phase of recovery is carried out when a resource manager of the group is connected to the relevant list structures. In an alternative embodiment, a more proactive recovery scheme is used which involves one or more resource managers within the group referring to the structure interest map to determine whether additional connections should be established. In this embodiment, if a failed resource manager performed uncommitted operations in relation to a list structure to which no other resource managers were connected, other resource managers within the group will establish a connection and perform the required recovery.

10

15

20